mitp

Sebastian Brabetz

Penetration Testing mit mimikatz

Das Praxis-Handbuch

Hacking-Angriffe verstehen und Pentests durchführen

Inhaltsverzeichnis

Vorwort

Ich hatte die Chance über das Aufbauen, Administrieren und Betreuen von Firewalls in einer größeren Firma in das Feld der IT-Security hineinzurutschen.

Beim täglichen Bearbeiten der Firewall-Regelwerke und Abschotten von Internet und DMZs gegenüber dem internen Netzwerk konnte ich ein gutes Gespür dafür entwickeln, was es bedeutet, Zugriffe möglichst einzugrenzen, aber auch dafür, Risiken in Form von freizugebenden Kommunikationskanälen gegen strikte IT-Security-Theorien abzuwägen.

Was mir das Administrieren von Firewalls allerdings nie vermitteln konnte, war eine verständliche Erklärung dafür, was Hacker wirklich tun und wie Angriffe auf IT-Systeme in der Realität aussehen.

Nach ein paar Jahren als Firewall-Administrator hatte ich die Chance, zwei Metasploit-Workshops von einem sehr talentierten Trainer beizuwohnen. Metasploit ermöglichte es mir, trotz fehlenden tiefgehenden Programmierhintergrunds zu verstehen, wie sich Software-Schwachstellen mittels Exploits ausnutzen lassen.

Seit diesen Metasploit-Workshops weiß ich besser zu schätzen, welche wichtige Aufgabe Firewalls erfüllen, indem sie nur die notwendigsten Dienste exponieren und Zugriffe auf das Nötigste beschränken können. Jedoch wurde mir auf der anderen Seite plötzlich auch bewusst, wie nutzlos Firewalls allein sind, wenn die Dienste, die man schlussendlich durch sie hindurch verfügbar machen will – und muss –, verwundbar sind.

Noch zwei weitere für meine Reise in der IT-Security wesentliche Erkenntnisse konnte ich aus diesen Metasploit-Workshops mitnehmen:

Erstens die Existenz des *Penetration Testing with Backtrack Linux* kurz PWB (mittlerweile *Penetration Testing with Kali Linux*, PWK) und der dazugehörigen OSCP-Zertifizierung, welche ich einige Jahre später auf Basis dieser beiden Workshops selbst absolviert habe.

Und zweitens die Existenz des Nessus-Schwachstellenscanners, den ich seitdem regelmäßig nutze, vertreibe und mit dessen Hilfe ich zum Thema Schwachstellenmanagement berate.

Neben dem Verständnis für Netzwerkkommunikation und deren Reglementierung hatte ich nun also auch ein gewisses Verständnis für Software-Schwachstellen, deren Ausnutzung sowie das systematische Auffinden und Vermeiden derselben.

Ein wichtiger Angriffsvektor, der mir weiterhin noch wenig geläufig war, stellten Konfigurationsschwachstellen dar, welche für sich allein genommen teilweise noch nicht mal unbedingt schlimm sein müssen. In Verbindung mit weiteren Zuständen in komplexen Firmennetzwerken können sie es aber ermöglichen, IT-Systeme und ganze IT-Landschaften zu kompromittieren.

Genau an dieser Stelle setzt mimikatz aus meiner Sicht als mächtiges Werkzeug an: mimikatz nutzt auf einer tiefen Ebene Möglichkeiten und Funktionen von Windows und den in Windows verwendeten Authentifizierungsprotokollen aus. Die richtigen (oder auch falschen) Personen können sich so trotz Firewalls, Virenscannern und Schwachstellenmanagement durch moderne Windows-Domänen bewegen wie Neo durch die Matrix.

Dieser letzte Vergleich ist sicherlich albern und ein Klischee, jedoch ist es dieser einfache Vergleich, mit dem ich diese Art von Schwachstellen und Angriffsvektoren für mich am besten greifbar machen und einordnen kann.

Sehr wichtig ist es mir, dass ich keinerlei Anerkennung für die in diesem Buch vorgestellten Programme und Angriffstechniken erlangen möchte. Alles, was in diesem Buch vorgestellt wird, wurde von sehr talentierten Menschen entwickelt und kostenlos dem Rest der Welt zur Verfügung gestellt, um transparent zu machen, welche Schwächen sich in Computersystemen befinden.

An dieser Stelle einzelne Namen zu nennen, wird wahrscheinlich der Tatsache nicht gerecht, dass auch diese Personen auf der Arbeit von anderen Personen vor ihnen aufgebaut haben. Insofern spare ich mir hier

das explizite nennen von Namen und verweise auf die Stellen im Buch, an denen ich auf die Personen oder Namen eingehe, die unmittelbar für die vorgestellten Programme oder Techniken eine Nennung verdienen.

Mit diesem Buch möchte ich das Wissen, welches ich mir über einen langen Zeitraum hart erarbeiten musste, anderen Personen leichter zugänglich machen, als es für mich zugänglich war.

Ich habe dabei auch keinerlei Angst, dass das Senken der Einstiegshürde in spannende IT-Security-Themen zu weniger Arbeit für mich oder andere IT-Security Professionals führen wird. Denn trotz stetiger Weiterentwicklung der Technik scheint eines derzeit auf der ganzen Welt nicht wirklich zu funktionieren: gänzlich sichere IT-Systeme und Programme zu entwickeln und aufzubauen.

Es herrscht ein Mangel an versiertem IT-Security-Personal und gleichzeitig werden Computer in immer mehr Bereichen des täglichen Lebens verankert: smarte Autos und Häuser, vernetzte Krankenhäuser, Personal-Fitness-Geräte und noch so vieles mehr.

Insofern ist dieses Buch für mich schon ein voller Erfolg, wenn nur eine einzige Person dadurch einen besseren Einblick in die Sicherheit von Windows-Domänen erlangt oder einfach nur Spaß an IT-Security hat.

Mein Beitrag an die IT-Security Community ist mit diesem Buch also primär das Absenken der Einstiegshürde in einen spannenden Bereich der IT-Security: Active Directory Security.

Abschließen möchte ich das Vorwort mit einem Dank an die Personen, die mir das Schreiben dieses Buches ermöglicht haben:

Uli

Der mitp-Verlag

Sabine Janatschek

Janina Bahlmann

Andrej Schwab

Martin Pizalla

Kapitel 1

Einleitung

1.1 Ziel und Inhalt des Buches

mimikatz hat wahrscheinlich jeder schon einmal gehört, der sich intensiver mit IT-Sicherheit auseinandersetzt. Über die Jahre hat sich mimikatz als eines der bekanntesten »Hacking-Tools« etabliert – nicht zuletzt als es für den Crypto-Trojaner NotPetya zweckentfremdet wurde, welcher in der zweiten Jahreshälfte 2017 um die Welt ging und unzählige Computer verschlüsselte.

Auch bei allen, die sich tiefgehend mit IT-Security auseinandersetzen, um z. B. Penetration Tester zu werden oder als Verteidiger ihre Unternehmen zu schützen, ist mimikatz schnell im Gespräch.

mimikatz ist vor allem für die Funktion bekannt, dem Arbeitsspeicher eines PCs, auf dem mimikatz läuft, Klartextpasswörter zu entlocken. Das ist nicht verwunderlich, da Klartextpasswörter, die am einfachsten zu verstehenden und weiterverwendbaren Geheimnisse darstellen, die man einem Computer entlocken kann.

Klartextpasswörter lassen sich ohne großes Verständnis dafür, wie Computersysteme und deren Sicherheitskonzepte funktionieren, weiterverwenden und beliebig an anderen Stellen ausprobieren. Nicht selten werden Passwörter für verschiedene Accounts, Dienste und Webseiten wiederverwendet, weshalb Klartextpasswörter oft zur Kompromittierung weiterer Daten und Systeme führen.

Auch liegt es in der Natur moderner Computersysteme, mittels sogenannter *Single-SignOn-Mechanismen* User automatisch und bequem in alle Dienste komplexer IT-Systemlandschaften einzuloggen. Diese Vertrauensstellungen zwischen Systemen führen dazu, dass man mit einem

Passwort nicht nur das System, von dem man es erhalten hat, kontrolliert, sondern auch unzählige weitere Ressourcen wie z. B. E-Mail-Konten, Webseiten und Kollaborationsplattformen wie SharePoint und viele weitere Ressourcen anzapfen und auslesen kann.

1.2 Mehr als nur Klartextpasswörter

All das ist sehr effektiv und in den falschen Händen auch schon sehr gefährlich – bzw. sehr hilfreich, wenn es von Verteidigern eingesetzt wird, um zielgerichtet Awareness zu schaffen und systematisch Sicherheitslücken aufzudecken. Allerdings kann mimikatz deutlich mehr als nur dem Arbeitsspeicher eines Windows-PCs Klartextpasswörter zu entlocken.

mimikatz ist quasi ein Maßgeschneidertes Tool, um die in Windows-Domänen eingesetzten Sicherheitsmechanismen und Protokolle wie z. B. NTLM und Kerberos gezielt auszunutzen und sich mit deren Hilfe durch Windows-Domänen zu hacken.

Wie Sie in späteren Kapiteln lesen werden, ist Kerberos keine Erfindung von Microsoft und findet auch abseits von Windows seine Anwendung. Nicht selten werden Linux- oder Mac-Systeme mithilfe von Kerberos in Windows-Domänen integriert. mimikatz kann also auch genutzt werden, um diese Geräte anzugreifen oder über diese Geräte den Rest einer Windows-Domäne anzugreifen.

Folglich stellt mimikatz ein umfangreiches Werkzeug dar, insbesondere zum Ausnutzen des Kerberos-Protokolls.

1.3 Zielgruppe des Buches und Vorraussetzungen zum Verständnis

Jeder, der sich mit IT-Sicherheit befasst, sollte wissen, wie einfach es ist, selbst den aktuellsten Windows-Versionen Passwörter zu entlocken. Für IT-Sicherheitsverantwortliche in Umgebungen mit Windows-Domänen sollte ein Verständnis von mimikatz und den damit möglichen Angriffen daher zur Pflichtlektüre gehören.

Mit diesem Buch möchte ich Ihnen einen leicht verständlichen Einstieg in die Funktionalität von mimikatz und Windows-Domänen-Eskalation geben. Natürlich können Sie die Funktionsweise von mimikatz auch im Internet recherchieren. Doch ich möchte Ihnen mit diesem Buch die komplexen Hintergründe des Programms zusammenhängend und verständlich näherbringen. Dabei setze ich nur grundlegende Kenntnisse im Bereich der IT-Security voraus, sodass dieses Buch sich sowohl an Einsteiger als auch an langjährige Profis richtet.

Nach einer kleinen Historie zu mimikatz werde ich Ihnen zuerst aufzeigen, wie Sie sich einfach eine kleine Testumgebung zum Nachspielen der Angriffe aufbauen können.

Danach werde ich gezielt auf einige Grundlagen in der Windows-Security-Architektur und auf das Kerberos-Protokoll eingehen, um die notwendigen Grundlagen für das Verständnis von mimikatz zu festigen.

Im Hauptteil des Buches werde ich dann gängige Angriffstechniken, die durch mimikatz ermöglicht werden, im Labor Schritt für Schritt erläutern, sodass Sie diese bei Bedarf gerne parallel durchspielen können.

Als kleines Highlight wird sich eines der Kapitel auch einer recht modernen Angriffstechnik – dem sogenannten Kerberoasting – widmen, welches zwar nun auch schon wieder seit ein paar Jahren bekannt, aber trotzdem noch nicht annähernd jeder Firma in Deutschland ein Begriff ist.

Um die vorgestellten Angriffe und Techniken in diesem Buch nachzuvollziehen und zu üben, benötigen Sie keinen Zugriff auf eine lebendige Firmenumgebung. Heutzutage ist es recht einfach möglich, mit kostenlosen Virtualisierungslösungen und kostenlosen Microsoft-Testinstallationen komplexe Windows-Domänen nachzustellen. Sie können problemlos alle Techniken in einer sicheren abgeschotteten Testumgebung erproben, ohne Gefahr zu laufen, die eigene Firma zu beeinträchtigen. Weiterhin können Sie problemlos, auch ohne Zugriff auf eine Firmenumgebung, wertvolles Know-How aufbauen und für den produktiven Einsatz erproben.

Zusammenfassend ist dieses Buch für jeden interessant, der noch kein mimikatz-Veteran ist und Interesse an IT-Security hat oder seinen Marktwert steigern möchte.

1.4 Rechtliches

Wahrscheinlich kommt kein Buch, welches sich um IT-Security dreht, ohne einen entsprechenden Warnhinweis aus: Das unbedarfte und unkontrollierte Anwenden von Werkzeugen wie mimikatz, kann (gegebenenfalls versehentlich) zu Straftaten führen. Es verstößt gegen deutsches Gesetz, IT-Systeme ohne Erlaubnis der Eigentümer auf Schwachstellen hin zu überprüfen oder gar Schwachstellen in diesen Systemen auszunutzen. Selbst mit Erlaubnis und Einverständniserklärung der Eigentümer kann es durchaus nicht rechtens sein, IT-Systeme zu auditieren. Nehmen wir einmal das Beispiel eines Mailservers in der eigenen Firma. Auf diesem Mailserver liegen gegebenenfalls vertrauliche oder private E-Mails, die dem deutschen Postgeheimnis entsprechend zu behandeln sind.

Auch Shared-Hosting-Umgebungen, wie sie z. B. bei jeglichen Cloud-Providern vorliegen, stellen ein Problem dar: Entdecken oder nutzen Sie gar eine Schwachstelle in der unterliegenden Infrastruktur des Cloud-Providers, so können Sie gegebenenfalls an Daten anderer Nutzer dieser Infrastruktur gelangen. Dies gilt es unbedingt zu vermeiden und bedarf ganz klarer vertraglicher Regelungen mit dem jeweiligen Provider.

Lassen Sie sich hiervon aber nicht abschrecken. Sicherheitsaudits sind auch in diesen Umgebungen sehr nützlich und wichtig. Gute Cloud-Provider lassen Sicherheitsaudits unter abgesteckten Bedingungen zu.

Auch könnte der Internet-Service-Provider, über dessen Infrastruktur ein einfacher Portscan durchgeführt werden soll, Portscans verbieten. Viele Internet-Service-Provider haben hierzu Klauseln in den Verträgen. Gerade bei privaten Anschlüssen wird das Portscanning gern pauschal verboten. Ich selbst habe zwar noch keine Fälle erlebt, bei denen Internet-Provider aufgrund des Verstoßes gegen dieses Verbot Anschlüsse gekündigt oder Kunden abgemahnt hätten, aber Sie gehen auf Nummer sicher, wenn Sie sich auch hier explizit eine Freigabe einholen.

Zu guter Letzt sollten Sie auch bedenken, dass es ein Kündigungsgrund sein kann, wenn Sie unbedarft mit mimikatz bei Ihrem Arbeitgeber experimentieren, selbst wenn Sie dabei nichts zerstören und nur gute Beweggründe haben.

Die Einverständniserklärung

Zu jedem Penetrationstest und Schwachstellenaudit gehört also immer eine schriftlich und vertraglich festgehaltene Einverständniserklärung des Eigentümers der Infrastruktur und aller beteiligten Provider. Vorlagen hierfür bekommen Sie beim Beauftragen von Schwachstellenscans und Penetrationstests bei professionellen Anbietern oder sicherlich auch frei verfügbar im Internet. Lassen Sie eine solche Vorlage aber vorsichtshalber durch Anwälte prüfen, bevor Sie größere Audits unternehmen.

IANAL – I am not a Lawyer

Dieses Buch stellt keine fundierte Rechtsberatung dar.

Ich möchte an dieser Stelle lediglich darauf hinweisen, dass die rechtlichen Rahmenbedingungen der IT-Security sehr ernst genommen werden müssen.

Im Zweifelsfall arbeiten Sie beim Lesen und Nachvollziehen dieses Buches komplett auf virtuellen Maschinen auf Ihrem privaten Computer oder besuchen entsprechend vorbereitete Workshops oder Weiterbildungen, die abgeschottete Demo-Umgebungen bereitstellen.

1.5 Begrifflichkeiten und Glossar

Zu guter Letzt möchte ich drauf hinweisen, dass es bei tiefgehenden Themen wie mimikatz und IT-Security immer mal wieder vorkommen kann, dass Ihnen einzelne Begriffe oder Hintergründe unklar sind. Ich habe daher versucht, entsprechende Begriffe direkt im Text durch **Fettschrift** kenntlich zu machen und im Glossar am Ende des Buches zu beschreiben.

Sollte Ihnen trotzdem beim Lesen noch etwas unklar sein, scheuen Sie sich nicht davor, den Begriff einfach in die Suchmaschine Ihrer Wahl einzugeben. Ich versichere Ihnen, dass Sie zu allen Inhalten in diesem Buch eine Vielzahl von Webseiten finden werden, die Ihnen die Hintergründe weiterführend erläutern.

Kapitel 6

Erste Schritte mit mimikatz

Bevor es mit mimikatz ans Eingemachte geht, sollten Sie die grundlegende Funktionalität, Syntax und Menüstruktur von mimikatz verstehen.

6.1 Vorbereiten von Windows für den ersten mimikatz-Start

Bevor Sie mimikatz herunterladen und starten können, müssen Sie sich erst einmal mit den wahrscheinlich unter Windows präsenten Virenscannern befassen.

6.1.1 Virenscanner: das Katz-und-Maus-Spiel

mimikatz ist bekannt wie ein bunter Hund! Jegliche Antivirenprogramm-Hersteller (kurz AV-Hersteller) wissen von der Existenz von mimikatz und haben teils bessere und teils schlechtere Erkennungspatterns für ihre Produkte erstellt, um mimikatz in den nackten unveränderten Versionen von Github zu erkennen.

In einem Pentest sind Sie natürlich mit Virenscannern konfrontiert. Sie müssen für die Verwendung von mimikatz auf einem Zielsystem in der anzugreifenden Domäne generell bereits Administrator sein, um die meisten Funktionalitäten von mimikatz nutzen zu können.

Als Administrator wäre es Ihnen z. B. möglich, den Virenscanner zu deinstallieren oder zu deaktivieren. Auch könnten Sie natürlich mit ein wenig Aufwand klassische AV-Produkte austricksen, indem Sie gezielt die Dinge im Code von mimikatz ändern, welche die AV-Scanner mit ihren limitierten Patterns alarmieren.

In der Vergangenheit reichten hierzu bereits einfache Techniken des Umbenennens von Strings im Quellcode (aus `mimikatz` wird `mimidogz`). Teilweise wurden auch kompliziertere Techniken, wie das Verschleiern oder Packen der Binärdateien eingesetzt, bis hin zum gänzlichen Neuimplementieren von mimikatz in anderen neuen Programmiersprachen wie z. B. Go, welche nicht von allen Virenscannern gut analysiert werden können.

Eines haben alle der vorgenannten Techniken aber gemein: Sobald sie für einen gewissen Zeitraum bekannt waren, haben sich Hersteller von klassischen AV-Produkten überlegt, wie sie diese Techniken identifizieren und abfangen können.

Genau hier liegt aber auch der Schwachpunkt von klassischen AV-Herstellern. Sie laufen immer nur einzelnen Techniken hinterher und schaffen neue limitierte Erkennungsmerkmale. Dadurch machen sie ihre Produkte immer invasiver und leistungshungriger, um einen vermeintlichen Schutz und damit gefühlte Sicherheit zu schaffen.

Verstehen Sie das bitte nicht falsch, ein Pentest wird deutlich spannender und aufwändiger, wenn auf allen Systemen, die man vorfindet, AV-Produkte installiert sind. Der Windows Defender z. B. ist im Bereich PowerShell-Erkennung sehr mächtig geworden, während sich z. B. Kaspersky sehr tief im System einnistet und schwer abzutöten ist. Allerdings hat bisher jede Pentest-Geschichte, die ich zu diesem Thema gehört habe, damit geendet, dass es mit irgendeiner neuen Technik schlussendlich dennoch möglich war, den AV-Scanner auszutricksen oder zu deaktivieren. In anderen Fällen haben die Pentester einfach solange gesucht, bis sie irgendwo ein auf der Domäne befindliches System vorgefunden haben, auf dem kein AV-Scanner installiert werden durfte oder versehentlich vergessen wurde.

Ein Lichtblick in Bezug auf die IT-Sicherheit sind neuere »Next-Gen-Antivirus-Produkte«, welche nicht mehr auf Basis von Bitmustern alarmieren, sondern mitunter leichtgewichtig im Betriebssystem bestimmte Funktionsaufrufe überwachen. Sie alarmieren gezielt auf eine Aneinanderreihung von Events im Betriebssystem und können so Prozesse beenden sowie Systeme abkapseln, noch bevor Schadsoftware oder mi-

mikatz ihren Dienst verrichten konnten. Aber auch diese Produkte sind wieder ein zusätzliches Level von Komplexität und sobald der Angreifer von ihrer Existenz weiß, kann er sich Mittel und Wege überlegen, sie zu umgehen oder entsprechend geschützte Systeme nicht weiter anzufassen.

Schlussendlich zielen alle präventiven Schutzsysteme wie klassische AV-Scanner oder verhaltensbasierende Produkte wie Next-Gen darauf ab, die erste Infektion zu verhindern oder zu alarmieren. Schafft es ein Angreifer erst einmal über vorgesehene Zugriffe in die Zielumgebung – z. B. über vom Benutzer ausgeführte, speziell für einen einzelnen Einsatz programmierte *Custom Malware* oder *legitim geklaute Zugangsdaten* oder meinetwegen über einen brandneuen, nicht bekannten *Zero-Day-Exploit* – so wird ein gut geschulter Angreifer früher oder später in der Lage sein, die komplette Umgebung zu übernehmen.

In diesem Buch werde ich das Thema AV-Evasion nicht weiter beleuchten. Wenn Sie Interesse haben, solche Techniken zu lernen und zu üben, so können Sie danach in der Suchmaschine Ihrer Wahl suchen und werden eine Vielzahl an Treffern finden. Erwarten Sie hierbei nicht eine einfache 1-Click-Lösung, die alle Virenscanner nachhaltig und für immer austrickst. Vielmehr basiert die Lösung zum Umgehen von Virenscannern immer darauf, dass man versucht gezielt den einen eingesetzten AV-Scanner über seine individuellen Schwächen auszutricksen. Hierzu wird der AV-Scanner in einem Labor installiert und nach dem Updaten von der Außenwelt abgeschnitten, sodass er den Hersteller nicht über Funde informieren kann.

Spannende Blogartikel und Anleitungen zu diesem Thema finden Sie regelmäßig bei der amerikanischen Pentesting-Firma *Black Hills Information Security* in englischer Sprache:

https://www.blackhillsinfosec.com/blog/

Halten Sie auch bei YouTube nach den jährlichen Videos mit dem Titel *Sacred CashCow Tipping* Ausschau, die ebenfalls von dieser Pentesting-Firma erstellt werden, um die AV-Industrie ein wenig in Bewegung zu halten.

6.1.2 Deaktivieren des Windows Defenders in der Laborumgebung

Wenn Sie Ihr eigenes Labor abweichend von dem im Buch vorgestellten Labor verwenden, so deaktivieren Sie ggf. einfach die verwendete AV-Lösung, sofern überhaupt eine installiert wurde.

Wenn Sie, wie in diesem Buch demonstriert, Windows Server 2016 installiert haben, so kommt dieser von Haus aus mit dem Windows Defender, welchen Sie über die Systemeinstellungen deaktivieren können.

Loggen Sie sich dazu als Administrator an dem Sprunghost ein und deaktivieren Sie alle Haken des Windows Defenders.

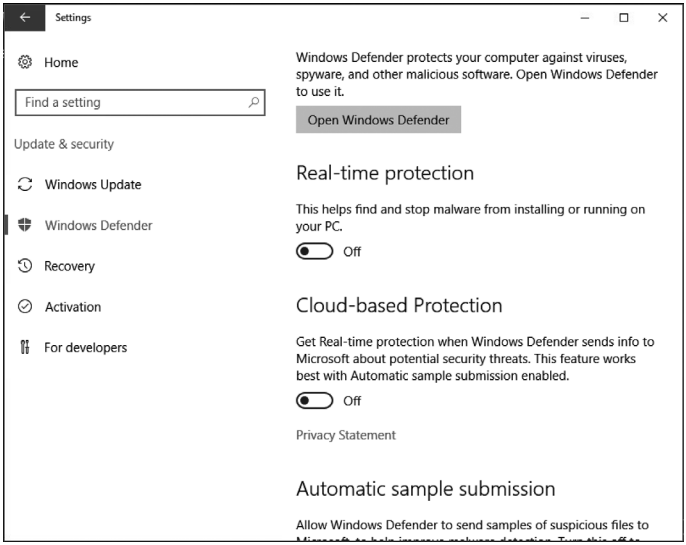

Abbildung 6.1: Deaktivieren des Windows Defenders

Deaktivieren Sie den Windows Defender vorerst auch wirklich nur auf dem Sprunghost und lassen Sie ihn auf dem Fileserver und dem Domänencontroller weiterlaufen. Stellen Sie dort auch gerne sicher, dass er mit allen aktuellen Updates versorgt ist. Dies wird Ihnen demonstrieren, dass ein schwaches Kettenglied oftmals ausreicht und es nahezu

fahrlässig ist, sich nur auf das Vorhandensein eines Virenscanners zu verlassen.

6.1.3 Herunterladen von mimikatz

Stellen Sie bitte sicher, dass Sie mimikatz stets aus dem offiziellen Github-Repository von Benjamin Delpy beziehen:

https://github.com/gentilkiwi/mimikatz

Das Herunterladen und Kompilieren des Quellcodes werde ich hier nicht weiter thematisieren, stattdessen werde ich der Einfachheit halber die vorkompilierten Binaries aus dem Repository verwenden, welche unter folgender Adresse zu finden sind:

https://github.com/gentilkiwi/mimikatz/releases

Spätestens dann, wenn Sie sich mit dem Thema AV-Evasion weiter befassen wollen, sollten Sie sich aber auch noch mal die Zeit nehmen und sich anschauen, wie Sie den Quellcode selbst kompilieren können. In Abbildung 6.2 sehen Sie die zum Zeitpunkt der Drucklegung des Buchs aktuellste Version von mimikatz 2.1.1.

Abbildung 6.2: Derzeit aktuelles Release von mimikatz 2.1.1

Nachdem Sie den Virenscanner als Administrator deaktiviert haben, loggen Sie sich bitte mit dem niedrig privilegierten Domänenbenutzer ein, welchen Sie im Zuge der Laborinstallation angelegt haben, und laden `mimkatz_trunk.zip` für diesen Benutzer herunter.

Sollten Sie beim Downloaden von mimikatz mit dem Internet Explorer Probleme mit der *Internet Explorer hardened Configuration* bekommen, so können Sie diese über den Server-Manager deaktivieren:

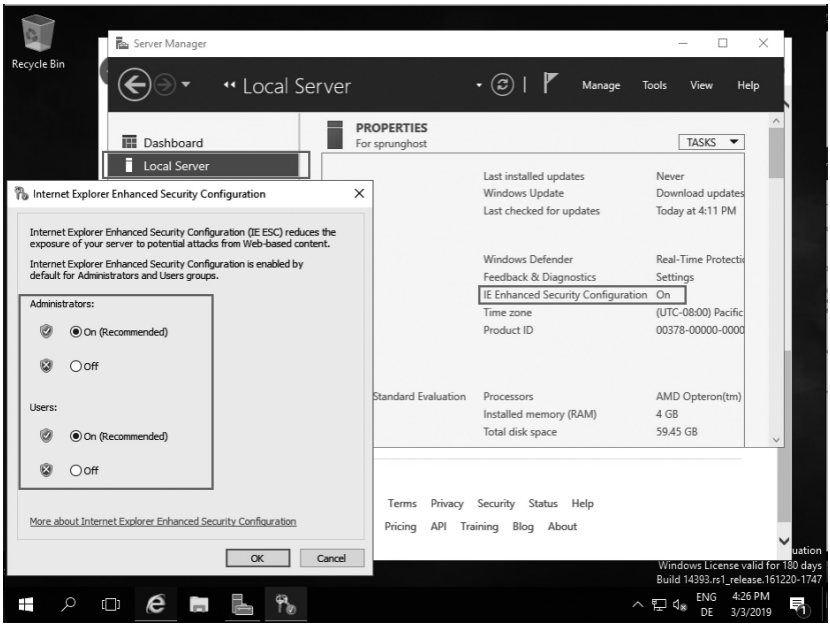

Abbildung 6.3: Deaktivieren der IE Enhanced Security Config.

Vergessen Sie nicht, ggf. den Internet Explorer neu zu starten, nachdem Sie die Deaktivierung durchgeführt haben. Spätestens dann sollte der Download mittels IE funktionieren.

6.1.4 Erste Start- und Gehversuche

Nach dem Herunterladen und Entpacken, können Sie die 64-Bit-Version von mimikatz aus dem Ordner x64 starten.

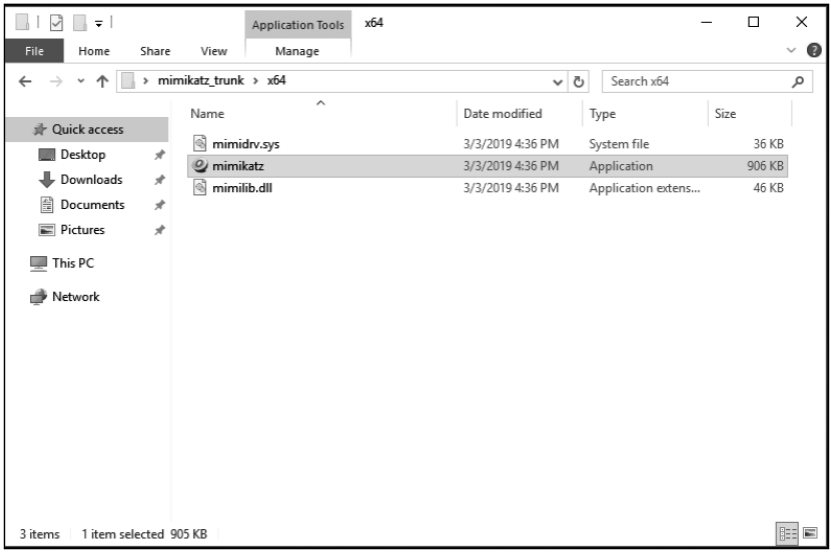

Abbildung 6.4: mimikatz X64 Version

Natürlich starten Sie auf einem 64-Bit-Betriebssystem eine 64-Bit-Version von mimikatz und auf einem 32-Bit-Betriebssystem die 32-Bit-Version. Sie können zwar auch die 32-Bit-Version von mimikatz auf einem 64-Bit-Windows starten, allerdings wird mimikatz aufgrund der WOW64-Abstrahierung dann nahezu nutzlos sein.

Nach dem Start werden Sie mit einem Banner und Versionsinformationen begrüßt.

Zur Syntax von mimikatz müssen Sie wissen, dass mimikatz nach Modulen strukturiert ist, welche abgegrenzt mit zwei Doppelpunkten Unterbefehle ermöglichen.

Um zu sehen, welche Module es alle gibt, können Sie einfach `::` (zwei Doppelpunkte) ohne jegliches Modul oder Befehlsnamen eingeben, wie in Abbildung 6.5 zu sehen ist. Daraufhin präsentiert mimikatz zuerst eine Fehlermeldung, welche sagt, dass das eingegebene Modul nicht gefunden werden konnte. Dies können Sie getrost ignorieren. Wichtiger ist die darauffolgende Liste an verfügbaren Modulen.

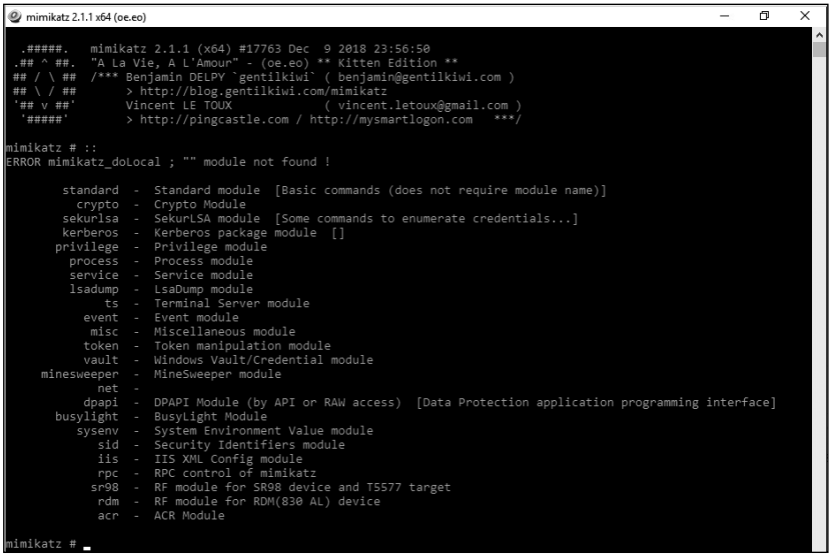

Abbildung 6.5: Erster Start von mimikatz und Anzeigen der Befehle.

Spätestens jetzt werden Sie den ein oder anderen Begriff aus den beiden Grundlagenkapiteln 4 und 5 wiederfinden, z. B.:

- `sekurlsa` – was eine absichtlich falsche Schreibweise von `secure` LSA darstellt – das Modul rund um die LSA
- `kerberos` – das Modul rund um die Interaktion mit Kerberos

Schauen Sie sich nun erst einmal das Modul `standard` an und prüfen, welche Funktionen sich hinter diesem Modul verstecken, indem Sie `standard::` eingeben.

Erneut startet die Ausgabe mit einer Fehlermeldung, dass der Befehl »null« nicht gefunden wurde, gefolgt von einer Übersicht aller Befehle, die sich hinter dem Standard-Modul befinden.

Stichwortverzeichnis